



UNITED STATES PATENT AND TRADEMARK OFFICE

mn
UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/699,629	10/30/2003	Matthew J. Burdick	MS1-1654US	3072

69316 7590 06/06/2007
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WA 98052

EXAMINER

BROWN, MICHAEL J

ART UNIT	PAPER NUMBER
----------	--------------

2116

MAIL DATE	DELIVERY MODE
-----------	---------------

06/06/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/699,629
Filing Date: October 30, 2003
Appellant(s): BURDICK ET AL.

MAILED

JUN 06 2007

Technology Center 2100

William J. Breen, III
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/5/2007 appealing from the Office action mailed 10/3/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2004/0243997	Mullen et al.
2002/0026436	Joory

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1, 6-9, 13-14, 18-19, 23-24, 28-30, 32-34, 36, and 40-41 are rejected under 35 U.S.C. 102(e) as being anticipated by Mullen et al.(US PGPub 2004/0243997).

As to claim 1, Mullen discloses a method comprising validating a configuration setting(preexisting application configuration settings 110, see Fig. 4) of a first application(preexisting application 108, see Fig. 4) for use with a second application(installed application 128, see Fig. 4), wherein the configuration setting including a first field(entry 80, see Fig. 3) and a first description of a first condition(file or parameter name, see paragraph 0029, lines 1-3) for the first field. Mullen also discloses the second application is composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having an attribute(application installation image 60, see Fig. 2), the attribute providing a second description of a second condition(file or

parameter name, see paragraph 0029, lines 1-3) for a second field(application configuration list 56, see Fig. 2), and the validating includes if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition, and if met, then determining that the configuration setting is valid for use with the second application(see paragraph 0029).

As to claim 5, Mullen discloses the method wherein the attribute is a declarative tag that that may be retrieved from and during execution of the second application, and does not determine the value of the field(see paragraph 0027, lines 5-13).

As to claim 6, Mullen discloses the method further comprising communicating a result of the validating to the first application(see paragraph 0029, lines 17-23).

As to claim 7, Mullen discloses the method wherein the configuration setting further comprises a field type(file, parameter or value, see paragraph 0023, line 18).

As to claim 8, Mullen discloses one or more computer-readable media(system installation package 50, see Fig. 2) comprising computer-executable instructions(installed application programs 4a, 4b....4n, see Fig. 1) that, when executed, perform the method as recited in claim 1.

As to claim 9, Mullen discloses a method comprising reading a first configuration setting(preexisting application configuration settings 110, see Fig. 4) of a first application(preexisting application 108, see Fig. 4) including a first field(entry 80, see Fig. 3) and a first description of a first condition(file or parameter name, see paragraph 0029, lines 1-3) for the first field, and examining a second application(installed

application 128, see Fig. 4) to find a second configuration setting(installed application configuration settings 130, see Fig. 4) that corresponds to the first configuration setting, wherein the second application is composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having an attribute(application installation image 60, see Fig. 2), the attribute providing a second description of a second condition(file or parameter name, see paragraph 0029, lines 1-3) for a second field(application configuration list 56, see Fig. 2), the second configuration setting having the second field and the second description. Mullen further discloses comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met then determining that the second configuration setting is valid for use with the first application(see paragraph 0029).

As to claim 13, Mullen discloses one or more computer-readable media(system installation package 50, see Fig. 2) comprising computer-executable instructions(installed application programs 4a, 4b....4n, see Fig. 1) that, when executed, perform the method as recited in claim 9.

As to claim 14, Mullen discloses a method comprising executing a documenter(file system 100, see Fig. 4) to find a plurality of fields(application configuration list 56, see Fig. 2) in an application(installed application 128, see Fig. 4), wherein the application is composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having attributes(application installation images 60, see Fig. 2), and each the attribute providing a description of a condition(file or

parameter name, see 0029, lines 1-3) for a respective the field. Mullen further discloses forming a configuration file(installed application configuration settings 130, see Fig. 4) having a plurality of configuration settings of the application, wherein each the configuration setting includes one the field and the description of the condition for the one the field, and outputting the configuration file, and utilizing the configuration file to validate configuration settings(see paragraph 0029).

As to claim 18, Mullen discloses one or more computer-readable media(system installation package 50, see Fig. 2) comprising computer-executable instructions(installed application programs 4a, 4b....4n, see Fig. 1) that perform the method as recited in claim 14.

As to claim 19, Mullen discloses a method comprising generating a configuration file(installed application configuration settings 130, see Fig. 4) having a plurality of configuration settings(preexisting application configuration settings 110, see Fig. 4) derived from a first application(preexisting application 108, see Fig. 4), wherein the application is composed of computer instructions(installation program 52, see Fig. 2), and the computer instructions having attributes(application installation images 60, see Fig. 2). Mullen further discloses each the attribute providing a description of a condition(file or parameter name, see paragraph 0029, lines 1-3) for a field(application configuration list 56, see Fig. 2), each the configuration setting having one the field and a corresponding the description, and validating whether the first application is valid for use with a second the application by comparing each the configuration setting of the first application with a corresponding the configuration setting of the second the

application to determine whether each the condition of the first application is met by a corresponding the condition of the second application(see paragraph 0029).

As to claim 23, Mullen discloses one or more computer-readable media(system installation package 50, see Fig. 2) comprising computer-executable instructions(installed application programs 4a, 4b...4n, see Fig. 1) that, when executed, perform the method as recited in claim 19.

As to claim 24, Mullen discloses a computer-readable medium comprising computer-executable instructions that, when executed by a computer(computer 2, see Fig. 4), direct the computer to read a first configuration setting(preexisting application configuration settings 110, see Fig. 4) of a first application(preexisting application 108, see Fig. 4) that includes a first field(entry 80, see Fig. 3) and a first description of a first condition(file or parameter name, see paragraph 0029, lines 1-3) for the first field, and validate whether the first condition is met by a second application(installed application 128, see Fig. 4), wherein the second application is composed of computer instructions(installation program 52, see Fig. 2). Mullen further discloses the computer instructions have an attribute(application installation images 60, see Fig. 2) that provides a second description of a second condition(file or parameter name, see paragraph 0029, lines 1-3) for a second field(application configuration list 56, see Fig. 2), and the first condition is validated through comparison with the second condition(see paragraph 0029).

As to claim 28, Mullen discloses a computer(computer 2, see Fig. 4) comprising a processor(processor 302, see Fig. 7), and memory(memory 304, see Fig. 7)

configured to maintain a first application(preexisting application 108, see Fig. 4) composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having an attribute(application installation images 60, see Fig. 2), the attribute providing a first description of a first condition(file or parameter names, see paragraph 0029, lines 1-3) for a first field(entry 80, see Fig. 3). Mullen further discloses a configuration file(file system 100, see Fig. 4) including a configuration setting(installed application configuration settings 130, see Fig. 4) of a second application(installed application 128, see Fig. 4) having a second field(application configuration list 56, see Fig. 2) and a second description of a second condition(file or parameter name, see paragraph 0029, lines 1-3) for the second field, and a configuration module that, when executed on the processor, validates the configuration setting for use with the first application by comparing the second description of the second condition with the first description of the first condition to determine whether the second condition is met by the first condition, and if met, then determining that the configuration setting is valid for use with the first application(see paragraph 0029).

As to claim 29, Mullen discloses the computer wherein the second application is stored in the memory(see paragraph 0025, lines 4-6 and paragraph 0035, lines 8-9).

As to claim 30, Mullen discloses the computer wherein the configuration file is received in a transmission from a network for storage in the memory(see paragraph 0024, lines 1-6 and paragraph 0035, lines 1-9).

As to claim 32, Mullen discloses a computer(computer 2, see Fig. 4) comprising a processor(processor 302, see Fig. 7), and memory(memory 304, see Fig. 7)

configured to maintain a first application(preexisting application 108, see Fig. 4) composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having an attribute(application installation images 60, see Fig. 2), the attribute providing a first description of a first condition(file or parameter name, see paragraph 0029, lines 1-3) for a first field(entry 80, see Fig. 3), and wherein a first configuration setting(preexisting application configuration settings 110, see Fig. 4) includes the first description and the first field. Mullen further discloses a configuration file(file system 100, see Fig. 4) including a second configuration setting(installed application configuration settings 130, see Fig. 4) of a second application(installed application 128, see Fig. 4) having a second field(application configuration list, 56, see Fig. 2) and a second description of a second condition(file or parameter name, see paragraph 0029, lines 1-3) for the second field, and a configuration module that, when executed on the processor, validates the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition, and if met, then determining that the first configuration setting is valid for use with the second application(see paragraph 0029).

As to claim 33, Mullen discloses the computer wherein the second application is stored in the memory(see paragraph 0025, lines 4-6 and paragraph 0035, lines 8-9).

As to claim 34, Mullen discloses the computer wherein the configuration file is received over a network and stored in the memory(see paragraph 0024, lines 1-6 and paragraph 0035, lines 1-9).

As to claim 36, Mullen discloses a content server(computer 2, see Fig. 4) comprising a broadcast transmitter(network card 308, see Fig. 7) configured to provide media content to a client in response to a request from the client, a processor(processor 302, see Fig. 7), and memory(memory 304, see Fig. 7) configured to maintain a first application(preexisting application 108, see Fig. 4) that when executed provides media content for broadcast by the broadcast transmitter, wherein the application is composed of computer instructions(installation program 52, see Fig. 2), the computer instructions have attributes(application installation images 60, see Fig. 2), and each the attribute provides a description of a condition(file or parameter name, see paragraph 0029, lines 1-3) for a field(entry 80, see Fig. 3). Mullen further discloses a documenter that is executable on the processor to generate a configuration file(system file 100, see Fig. 4) having a configuration setting(preexisting application configuration settings 110, see Fig. 4) of the first application, wherein the configuration setting includes the field(application configuration list 56, see Fig. 2) and the description of the condition(file or parameter name, see paragraph 0029, lines 1-3) for the field.

As to claim 40, Mullen discloses a content server(computer 2, see Fig. 4) comprising a first application(preexisting application 108, see Fig. 4) composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having attributes(application installation images 60, see Fig. 2), and each the attribute providing a description of a condition(file or parameter name, see paragraph 0029, lines 1-3) for a field(entry 80, see Fig. 3). Mullen further discloses a configuration module that is executable to validate whether each the condition is met by a second

application(installed application 128, see Fig. 4), and a documenter that is executable to generate a configuration file(system file 100, see Fig. 4) having a configuration setting(preexisting application configuration settings 110, see Fig. 4) of the first application, wherein the configuration setting includes the field and the description of the condition.

As to claim 41, Mullen discloses the content server further comprising a broadcast transmitter(network card 308, see Fig. 7), wherein the first application, when executed, provides content for broadcast by the broadcast transmitter.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 2-4, 10-12, 15-17, 20-22, 25-27, 31, 35, 37-39, and 42-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mullen et al.(US PGPub 2004/0243997) in view of Joory(US PGPub 2002/0026436).

As to claim 2, Mullen discloses the method as cited in claim 1 and explained above, however Mullen fails to disclose the method wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a method wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a sting pattern, cardinality of a collection, a mandatory value, and an optional value(see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a method where the first and second conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 3, Joory discloses a method wherein the first and second conditions are default values(see paragraph 0040, lines 5-6).

As to claim 4, Joory teaches a method wherein the first and second conditions are textual descriptions selected from the group consisting of a description of units in which the respective first and second fields are expressed, a description of meanings of the respective first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37).

As to claim 10, Mullen discloses the method as cited in claim 9 and explained above, however Mullen fails to disclose the method wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a method wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a sting pattern, cardinality of a collection, a mandatory value, and an optional value(see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a method where the first and second conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 11, Joory teaches a method wherein the first and second conditions are default values(see paragraph 0040, lines 5-6).

As to claim 12, Joory teaches a method wherein the first and second conditions are textual descriptions selected from the group consisting of a description of units in which the respective first and second fields are expressed, a description of meanings of the respective first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37).

As to claim 15, Mullen discloses the method as cited in claim 14 and explained above, however Mullen fails to disclose the method wherein the condition is a value constraint selected from the group consisting of an integer range, a float range, a value

Art Unit: 2116

set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a method wherein the condition is a value constraint selected from the group consisting of an integer range, a float range, a value set, a sting pattern, cardinality of a collection, a mandatory value, and an optional value (see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a method where the conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 16, Joory teaches a method wherein the condition is a default value(see paragraph 0040, lines 5-6).

As to claim 17, Joory teaches a method wherein the condition is a textual description selected from the group consisting of a description of a unit in which the respective the field is expressed, a description of meanings of the respective the field, and an example of a value of the respective the field(see paragraph 0049, lines 9-37).

As to claim 20, Mullen discloses the method as cited in claim 19 and explained above, however Mullen fails to disclose the method wherein each condition is a value constraint selected from the group consisting of an integer range, a float range, a value set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a method wherein each condition is a value constraint selected from the group consisting of an integer range, a float range, a value set, a sting pattern,

cardinality of a collection, a mandatory value, and an optional value (see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a method where each condition is compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 21, Joory teaches a method wherein each condition is a default value(see paragraph 0040, lines 5-6).

As to claim 22, Joory teaches a method wherein each condition is a textual description selected from the group consisting of a unit in which a corresponding the field is expressed, description of a meanings of a corresponding the field, and an example of a value a corresponding the field(see paragraph 0049, lines 9-37).

As to claim 25, Mullen discloses the computer-readable medium as cited in claim 24 and explained above, however Mullen fails to disclose the method wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a computer-readable medium wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a sting pattern, cardinality of a collection, a mandatory value, and an optional value (see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a computer-readable medium where the first and second conditions are

compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 26, Joory teaches a computer-readable medium wherein the first and second conditions are default values(see paragraph 0040, lines 5-6).

As to claim 27, Joory teaches a computer-readable medium wherein the first and second conditions are textual descriptions selected from the group consisting of a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37).

As to claim 31, Mullen discloses the computer as cited in claim 28 and explained above, however Mullen fails to disclose the computer wherein the first and second conditions are selected from the group consisting of a value constraint, a default value, a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields.

Joory teaches a computer wherein the first and second conditions are selected from the group consisting of a value constraint, a default value, a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a computer where

the first and second conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 35, Mullen discloses the computer as cited in claim 32 and explained above, however Mullen fails to disclose the computer wherein the first and second conditions are selected from the group consisting of a value constraint, a default value, a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields.

Joory teaches a computer wherein the first and second conditions are selected from the group consisting of a value constraint, a default value, a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a computer where the first and second conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 37, Mullen discloses the content server as cited in claim 36 and explained above, however Mullen fails to disclose the content server wherein the condition is value constraint selected from the group consisting of an integer range, a float range, a value set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a content server wherein the first and second conditions are value constraints selected from the group consisting of an integer range, a float range, a value set, a sting pattern, cardinality of a collection, a mandatory value, and an optional value(see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a content server where the first and second conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 38, Joory teaches a content server wherein the condition is a default value(see paragraph 0040, lines 5-6).

As to claim 39, Joory teaches a content server wherein the condition is a textual description selected from the group consisting of, a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37).

As to claim 42 Mullen discloses the content server as cited in claim 40 and explained above, however Mullen fails to disclose the content server wherein each the condition is a value constraint selected from the group consisting of an integer range, a float range, a value set, a string pattern, cardinality of a collection, a mandatory value, and an optional value.

Joory teaches a content server wherein each the condition is a value constraint selected from the group consisting of an integer range, a float range, a value set, a sting

pattern, cardinality of a collection, a mandatory value, and an optional value(see paragraph 0049, lines 9-37). It would have been obvious to one of ordinary skill in the art to combine the inventions of Mullen and Joory in order to create a content server where the first and second conditions are compared in more detail. The motivation to do so would be to assure dependable validation of transferring configuration settings.

As to claim 43, Joory teaches a content server wherein each the condition is a default value(see paragraph 0040, lines 5-6).

As to claim 44, Joory discloses a content server wherein each the condition is a textual description selected from the group consisting of a description of a unit in which a respective the field is expressed, a description of a meaning of a respective the field, and an example of a values of a respective the field(see paragraph 0049, lines 9-37).

As to claim 45, Mullen discloses a system comprising a first computer(computer 2, see Fig. 4) communicatively coupled to the network and including a first application(preexisting application 108, see Fig. 4) composed of computer instructions(installation program 52, see Fig. 2), the computer instructions having a first attribute(application installation images 60, see Fig. 2) that provides a first description of a first condition(file or parameter name, see paragraph 0029, lines 1-3) for the first field(entry 80, see Fig. 3), wherein a first configuration setting(preexisting application configuration settings 110, see Fig. 4) includes the first field and the first description, a second application(installed application 128, see Fig. 4) composed of computer instructions(installation program 52, see Fig. 2) having a second attribute(application installation images 60, see Fig. 2) that provides a second description of a second

condition(file or parameter name, see paragraph 0029, lines 1-3) for a second field(application configuration list 56, see Fig. 2), and a configuration module that is executable by the second computer to validate the first configuration setting for use with the second application by comparing the first description of the first condition with the second description of the second condition(see paragraph 0029). However Mullen fails to disclose the system comprising a network, a first computer, and a second computer communicatively coupled to the network and including the second application.

Joory teaches a system comprising a network(network 260, see Fig. 2), a first computer(system 205, see Fig. 2), and a second computer(system 210, see Fig. 2) all communicatively couple to one another. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions of Mullen and Joory in order to create a multi-computer system that can transfer configuration settings between applications. The motivation to do so would be to be able to validate configuration settings between applications from one computer to the other.

As to claim 46, Joory teaches a system wherein the first computer is configured as a set-top box and the second computer is configured as a content server(see paragraph 0037).

As to claim 47, Joory teaches a system wherein the first and second conditions are selected from the group consisting of a value constraint, a default value, a description of units in which the first and second fields are expressed, a description of meanings of the first and second fields, and an example of values of the first and second fields(see paragraph 0049, lines 9-37).

(10) Response to Argument

A) First Ground of Rejection: *Claims 1, 6-9, 13-14, 18-19, 23-24, 28-30, 32-34, 36, and 40-41 satisfy the requirements of 35 U.S.C. § 102(e) and therefore are not anticipated by Mullen.*

I. *Appellant argues specifically that Mullen does not disclose validating including “if the first field corresponds to the second field, then comparing the first description of the first condition with the second description of the second condition to determine whether the first condition is met by the second condition” or “if met, then determining that the configuration setting is valid for use with the second application”.*

Examiner continues to disagree as Mullen discloses that if the entry 80 corresponds to the application configuration list 56, then comparing the files or parameters or values(see paragraph 0023, line 18) to determine whether the configuration setting is valid for use with the installed application 128(see paragraph 0023, lines 10-18).

Appellant continues to argue that Examiner’s previous response makes no mention of a condition for a field nor a comparison of these conditions.

Examiner disagrees as the condition(file or parameter name; see paragraph 0029, line 3) for the field(configuration setting entry 80; see paragraph 0029, lines 1-2)

is mentioned. Moreover, a comparison of these conditions can be inferred from the statement "If the configuration settings to copy comprise parameters in a data structure, such as a registry file or other file, then the....."(see paragraph 0029, lines 17-23).

II. *Appellant argues specifically that Mullen fails to teach an attribute being a "declarative tag".*

Examiner continues to disagree as Mullen discloses application installation image 60 able to be retrieved from and during execution of the installed application 128(see paragraph 0027, lines 5-17). Clearly the installation image 60 performs the functionality of the claimed "declarative tag".

III. *Appellant argues specifically that Mullen is incapable of "communicating a result" of the validation.*

Examiner disagrees as Mullen communicates a result when "the installation program 52 opens the data structure to access the configuration settings to copy, and then opens the corresponding data structure in the new installation to update with the copied configuration setting value.

IV. *Appellant argues specifically that Mullen does not examine a second application.*

Examiner continues to disagree as Mullen discloses examining the installed application(see paragraph 0023, lines 10-17). The second application is examined

when the installation program 52 finds the second configuration setting(installed application configuration settings 130; paragraph 0024, lines 13-18).

V. *Appellant argues specifically that Mullen fails to teach or disclose computer readable media having the performance capability of claim 14(validate configuration settings).*

Examiner disagrees as Mullen discloses a system installation package 50, which carries out the validation of configuration settings as set forth in claim 14(see paragraph 0023, lines 1-17). For further clarification the installation package 50 includes installation program 52, which carries out the validation of the configuration settings(see paragraph 0028, lines 4-9).

VI. *Appellant argues specifically that Mullen fails to teach computer-readable medium which when executed causes a computer to “validate whether the first condition is met by a second application.....”*

Examiner continues to disagree as Mullen discloses a system installation package 50, which validates whether a pre-existing application is validated by running the installed second application(see paragraph 0023, lines 1-17). Again the installation package 50 includes installation program 52, which carries out the validation of conditions(see paragraph 0028, lines 4-9).

VII. *Appellant argues specifically that Mullen fails to disclose a content server that includes a broadcast transmitter or memory.*

Examiner continues to disagree as Mullen discloses a computer 2, which includes a network card 308 and memory 304(see Fig. 7 and paragraph 0035). The network card 308 serves as a broadcast transmitter in that it transmits data via an Ethernet network. It is also evident that the memory 304 within the content server(computer 300) satisfies the content servers inclusion of memory.

B) Second Ground of Rejection: *Claims 2-4, 10-12, 15-17, 20-22, 25-27, 31, 35, 37-39, and 42-47 satisfy the requirements of 35 U.S.C. § 103(a) and therefore are not unpatentable over Mullen in view of Joory.*

I. *Appellant argues specifically that Joory fails to teach “wherein the first and second conditions are value constraints.”*

Examiner disagrees as Joory introduces conditions(attributes; see paragraph 0049, line 10) that are value constraints(server names, mailbox, password, etc.....; see paragraph 0049, lines 12-20).

II. *Appellant argues specifically that Joory fails to teach “a value constraint selected from the group consisting of: an integer range; a float range; a value set;*

a string pattern; cardinality of a collection; a mandatory value; and an optional value."

Examiner disagrees as Joory introduces the attributes being selected from a group consisting of current view, "customized current view" settings, "define view" settings, etc...(see paragraph 0049, lines 25-32).

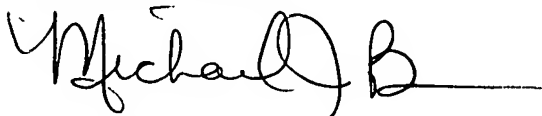
(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

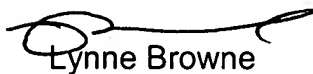
For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Michael J. Brown

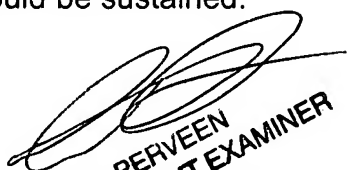


Conferees:

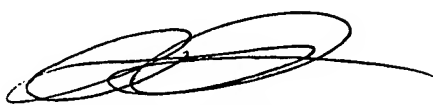


Lynne Browne

Rehana Perveen



REHANA PERVEEN
SUPERVISORY PATENT EXAMINER
6/4/07



REHANA PERVEEN
SUPERVISORY PATENT EXAMINER
6/4/07